

**We claim:**

1. A multiprocessor system, comprising:
  - a plurality of processor modules, including a software management processor;
  - a non-volatile storage memory configuration (NVS) coupled to the software management processor; and
  - means for uploading and downloading system software and data between the processor modules and the NVS, whereby only the software management processor has read or write access to the NVS.
2. The multiprocessor system of claim 1, wherein a software mechanism operates on the software management processor to control the transfer of software or data between the NVS and the processor modules.
3. The multiprocessor system of claim 2, wherein the software mechanism comprises a DHCP server.
4. The multiprocessor system of claim 2, wherein the software mechanism comprises an FTP server.
5. The multiprocessor system of claim 1, wherein the NVS is coupled to the software management processor by a dedicated storage device access bus.

- 
6. The multiprocessor system of claim 1, wherein the processor modules are coupled together by a communication bus.
  7. The multiprocessor system of claim 6, wherein the processor modules are physically coupled to a backplane that provides the communication bus.
  8. The multiprocessor system of claim 1, wherein the NVS comprises two redundant storage devices.
  9. The multiprocessor system of claim 8, wherein the two redundant storage devices are non-volatile memory devices.
  10. The multiprocessor system of claim 1, wherein the uploading and downloading means is a software version management module (SVM) that is executed by the software management processor and controls read and write access to the NVS.
  11. The multiprocessor system of claim 1, wherein the multiprocessor system is configured as a node in a ring network.
  12. The multiprocessor system of claim 11, wherein the ring network is a synchronous optical network.

13. The multiprocessor system of claim 11, wherein the processor modules operate as either traffic carrying modules or cross-connect modules for the ring network.
14. A multiprocessor system, comprising:
- a plurality of processor modules including a software management processor;
  - a non-volatile storage memory configuration (NVS) coupled to the software management processor, and having a plurality of redundant storage devices that store redundant copies of system software; and
  - a software version management module (SVM) executed by the software management processor, that manages the system software stored in the NVS, controls read and write access between the NVS and the software management processor, and enables system software to be loaded from the software management processor to the processor modules.
15. The multiprocessor system of claim 14, wherein the SVM prevents the redundant storage devices from being accessed simultaneously.
16. The multiprocessor system of claim 14, wherein each redundant storage device has a file system comprising:
- a current context area containing a copy of system software that is accessible to the SVM for upload to the processor modules; and
  - an alternate context area that is accessible to the SVM for downloading a different version of system software.

17. The multiprocessor system of claim 16, wherein the alternate context area and current context area in each redundant storage device may be switched, whereby system software in the alternate context area becomes accessible to the SVM for upload to the software management processor.
18. The multiprocessor system of claim 17, wherein the alternate context area and current context area in each redundant storage device are switched by the SVM at system initialization.
19. The multiprocessor system of claim 16, wherein the current and alternate context areas in each redundant storage device include a plurality of component files that store system software components needed for the processor modules to perform their functionality.
20. The multiprocessor system of claim 19, wherein the current and alternate context areas in each redundant storage device further include a software generic control (SGC) file that stores data used to match the system software components with one or more of the processor modules.
21. The multiprocessor system of claim 20, wherein a checksum is associated with each system software component, and the SVM uses the checksums to validate the integrity of the system software.
22. The multiprocessor system of claim 21, wherein the checksum is stored in the software generic control file.

23. The multiprocessor system of claim 16, wherein the current and alternate context areas in each redundant storage device include a configuration file.
24. The multiprocessor system of claim 23, wherein a checksum is associated with each configuration file, and the SVM uses the checksum to validate the integrity of the configuration file.
25. The multiprocessor system of claim 14, wherein the NVS comprises a primary NVS device and a secondary NVS device that respectively store a designated and backup copy of the system software.
26. The multiprocessor system of claim 25, wherein the primary NVS device and secondary NVS device designations do not have a permanent relationship with a specific redundant storage device, and may be assigned new designations by the SVM.
27. The multiprocessor system of claim 25, wherein the system software is organized in a flash file system comprising:
- a primary current context area in the primary NVS device that stores the designated copy of system software which is accessible to the SVM for upload to the processor modules through the software management processor;
  - a secondary current context area in the secondary NVS device that stores the backup copy of system software which is accessible to the SVM for upload to the processor modules through the software management processor;

a primary alternate context area in the primary NVS device that is accessible to the SVM for downloading a different version of system software; and

a secondary alternate context area in the second primary NVS device that is accessible to the SVM to backup the different version of system software.

28. A communication system, comprising:

a plurality of multiprocessor systems coupled in a ring network, and comprising,

    a plurality of processor modules coupled together that operate in the ring network as either traffic carrying modules or cross-connect modules,

        a software management processor,

        a non-volatile storage memory configuration (NVS) coupled to the software management processor, and having a plurality of redundant storage devices that store redundant copies of system software, and

            a software version management module (SVM) executed by the software management processor, that manages the system software stored in the NVS, controls read and write access between the NVS and the software management processor, and enables system software to be loaded from the software management processor to the processor modules; and

    a plurality of personal computers coupled to each multiprocessor system in the ring network.

29. The communication system of claim 28, wherein the plurality of personal computers are coupled to each multiprocessor system through a LAN router.

30. The communication system of claim 28, wherein the plurality of personal computers are coupled to each multiprocessor system through an ATM switch.
31. The communication system of claim 28, wherein a portion of the plurality of personal computers are coupled to each multiprocessor system through a LAN router, and another portion of the plurality of personal computers are coupled to each multiprocessor system through an ATM switch.
32. The communication system of claim 28, wherein the ring network is a synchronous optical network.
33. A method of managing system software in a multiprocessor system having a plurality of processor modules and a plurality of non-volatile storage devices, comprising the steps of:  
storing a redundant copy of the system software in each non-volatile storage device;  
restricting read and write access to the plurality of non-volatile storage devices to a software management processor that is coupled to the plurality of processor modules; and  
loading the system software to the plurality of processor modules by retrieving the system software with the software management processor, and then loading the system software through the software management processor to the plurality of processor modules.
34. The method of claim 33, wherein a new version of system software may be stored in the plurality of non-volatile storage devices by the additional steps of:

- downloading the new version of system software from the software management processor to one of the non-volatile storage devices; and
- copying the new version of system software to each additional non-volatile storage device.
35. The method of claim 33, wherein the plurality of redundant storage devices comprise a primary NVS device and a secondary NVS device that respectively store a designated and a backup copy of the system software.
36. The method of claim 34, wherein the new version of system software is downloaded to a primary NVS device, and copied from the primary NVS device to a secondary NVS device.
37. The method of claim 36, wherein access to the primary NVS device is restricted while the new version of system software is being copied to the secondary NVS device.
38. The method of claim 35, wherein the primary and secondary NVS devices each include a current context area and an alternate context area, and the system software is loaded to the software management processor from the current context area.
39. A method of performing a system upgrade in a multiprocessor system having a primary non-volatile storage (NVS) devices, a secondary NVS device and a plurality of processor modules, comprising the steps of:

identifying component files in the primary NVS device that store system software components used by the multiprocessor system;

downloading a new version of system software through a software management processor to the identified component files in the primary NVS device;

creating a backup copy of the new version of system software in a secondary NVS device by copying the identified component files from the primary NVS device;

loading the new version of system software from the primary NVS device to the plurality of processor modules through the software management processor.

40. The method of claim 39, wherein the new version of system software is downloaded through the software management processor using an FTP server operating on the software management processor.
41. The method of claim 39, wherein the new version of system software is downloaded through the software management processor using an FTAM server operating on the software management processor.
42. The method of claim 39, wherein the step of identifying component files in the primary NVS device that store system software components used by the multiprocessor system is performed by receiving a package file at the software management processor that includes a list of the component files to be upgraded.

43. The method of claim 42, comprising the additional step of:  
receiving a system command at the software management processor that includes file transfer parameters necessary to retrieve the package file.
44. The method of claim 39, wherein the step of loading the new version of system software from the primary NVS device to the software management processor is preceded by the step of:  
initiating a system wide initialization command.
45. The method of claim 39, comprising the additional step of:  
generating an autonomous output message if the system upgrade requires modifications to a programmable device on any processor module.
46. The method of claim 39, wherein the multiprocessor system is configured as a node in a ring network, and the system upgrade is performed remotely.
47. A method of performing a partial system upgrade in a multiprocessor system having a non-volatile storage memory configuration (NVS) and a plurality of processor modules, comprising the steps of:  
identifying component files in a primary NVS device that store system software components used by the processor modules to be upgraded;  
downloading a new version of system software through a software management processor to the identified component files in the primary NVS device;

F00000000000000000000000000000000

creating a backup copy of the new version of system software in a secondary NVS device by copying the identified component files from the primary NVS device; loading the new version of system software from the primary NVS device to the plurality of processor module through the software management processor.

48. The method of claim 47, wherein the step of loading the new version of system software is preceded by the additional step of:

copying any component files other than the identified component files from the primary NVS device to the secondary NVS device.

49. The method of claim 47, wherein the component files in the primary NVS device are identified by the steps of:

receiving a new software generic control (SGC) file; and  
comparing the new SGC file with a previous SGC file stored on the NVS.

50. The method of claim 47, wherein the step of loading the new version of system software from the primary NVS device to the software management processor is preceded by the step of:

initiating an initialization command that acts only on the processor modules to be upgraded.

51. The method of claim 47, comprising the additional step of:

generating an autonomous output message if the partial system upgrade requires modifications to a programmable device on any processor module.

52. The method of claim 47, wherein the multiprocessor system is configured as a node in a ring network, and the partial system upgrade is performed remotely.

53. A method of managing processor configurations in a multiprocessor system having a non-volatile storage memory configuration (NVS) and a plurality of processor modules, comprising the steps of:

storing a backup configuration in the NVS;

checking for a configuration file in the NVS;

if the configuration file is found in the NVS, then loading the configuration file to the plurality of processor modules through a software management processor;

if the configuration file is not found in the NVS, then determining whether the backup configuration is supported by the multiprocessor system;

if the configuration file is not found in the NVS and the backup configuration is supported by the multiprocessor system, then loading the backup configuration to the plurality of processor modules through a software management processor.

54. The method of claim 53, wherein if the configuration file is not found in the NVS and the backup configuration is not supported by the multiprocessor system, then providing a user with an option to restore a factory default configuration.

55. The method of claim 53, wherein a software version management module automatically stores the backup configuration.

56. The method of claim 53, wherein the step of checking for a configuration file is performed by the software management processor when the multiprocessor system is initialized.

57. A method of synchronizing access to a non-volatile storage memory configuration (NVS) in a multiprocessor system, comprising the steps of:

receiving a file operation request;

checking if a semaphore is active;

if the semaphore is active, then blocking access to the NVS until a previous file operation is complete and the semaphore is deactivated;

if the semaphore is not active, then granting access to a primary NVS device and activating the semaphore;

performing a file operation on the primary NVS device;

restricting access to the primary NVS device and granting access to a secondary NVS device;

performing a backup of the file operation on the secondary NVS device; and

deactivating the semaphore.

58. The method of claim 57, comprising the additional steps of:

assigning a transaction ID when access to the primary NVS device is granted; and

performing additional file operations on the primary NVS device upon receiving additional file operation requests including the transaction ID.

59. The method of claim 57, wherein the step of restricting access to the primary NVS device and granting access to a secondary NVS device is preceded by the step of:

receiving an command indicating that the file operation on the primary NVS device is complete, and requesting access to the secondary NVS device.

60. The method of claim 57, wherein a first semaphore is used to gain read access to a current context area on the primary and secondary NVS devices, and a second semaphore is used to gain write access to an alternate context area on the primary and secondary NVS devices.

61. The method of claim 60, wherein a third semaphore ensures that only one context area is accessed at any given time.

62. A method of activating a new software version in a multiprocessor system having a redundant flash file system (FFS) with a current context area and an alternate context area, comprising the steps of:

loading the new software version in the alternate context area;

receiving a context switch command;

setting an alternate boot flag that instructs the multiprocessor system to load the new software version from the alternate context area upon system initialization;

loading the new software version to the multiprocessor system from the alternate context area upon system initialization;

clearing the alternate boot flag, whereby the multiprocessor system returns to its ordinary state of loading software from the current context area upon system initialization;

validating that the new software has loaded and is functioning properly;  
if the new software has not loaded or is not functioning properly, then loading a previous software version to the multiprocessor system from the current context area; and  
if the new software has loaded and is functioning properly, then activating the alternate context area, whereby the alternate context area is redesignated as the current context area.

63. The method of claim 62, wherein a designated copy of the FSS is located on a primary NVS device and a backup copy of the FFS is located on a secondary NVS device.

64. The method of claim 62, wherein if the new software has loaded and is functioning properly, then the new software version in the alternate context area is designated as the current context area.

65. A method of initializing a multiprocessor system having a non-volatile storage memory configuration (NVS) and a plurality of processor modules, comprising the steps of:

verifying the integrity of system software stored in a primary NVS device;

if the system software stored in the primary NVS device is valid, then loading the system software from the primary NVS device to a software management processor;

if the system software stored in the primary NVS device is corrupt, then accessing a secondary NVS device and verifying the integrity of a backup copy of system software;

if the secondary NVS device has been accessed and the backup copy of system software is valid, then loading the backup copy of system software from the secondary NVS device to the software management processor;

if the secondary NVS device has been accessed and the backup copy of system software is corrupt, then replacing the primary NVS device and returning to the step of verifying the integrity of system software stored in the primary NVS system; and

loading the system software or backup copy of the system software from the software management processor to the plurality of processor modules.

66. The method of claim 65, wherein if the secondary NVS device has been accessed and the backup copy of system software is corrupt, then downloading a new copy of system software to the primary NVS device and returning to the step of verifying the integrity of system software stored in the primary NVS device.

67. A method of initializing a multiprocessor system having a non-volatile storage memory configuration (NVS) and a plurality of processors, comprising the steps of:

verifying the integrity of system software stored in a designated NVS device;

if the system software stored in the designated NVS device is valid, then assigning the designated NVS device as a primary NVS device;

if the system software stored in the designated NVS device is corrupt, then accessing a backup NVS device and verifying the integrity of a backup copy of system software;

if the backup NVS device has been accessed and the backup copy of system software is valid, then assigning the backup NVS device as the primary NVS device;

if the backup NVS device has been accessed and the backup copy of system software is corrupt, then replacing the designated NVS device and returning to the step of verifying the integrity of system software stored in a designated NVS device;

loading system software from the primary NVS device to a software management processor; and

loading system software from the software management processor to the plurality of processors.

68. The method of claim 67, wherein if the backup NVS device has been accessed and the backup copy of system software is corrupt, then downloading a new copy of system software to the designated NVS device and returning to the step of verifying the integrity of system software in a designated NVS device.

69. The method of claim 67, comprising the additional steps of:

- assigning the backup NVS device as a secondary NVS device if the system software stored in the designated NVS device is valid; and
- assigning the designated NVS device as the secondary NVS device if the backup NVS device has been accessed and the backup copy of system software is valid.

70. The method of claim 67, wherein the step of verifying the integrity of system software in a designated NVS device is preceded by the step of:

- initiating a system wide initialization command on the software management processor.

71. A multiprocessor system, comprising:

- a non-volatile storage memory configuration (NVS) having a primary NVS device and a secondary NVS device;

a software management processor having exclusive read and write access to the NVS; a software version management module (SVM) executed by the software management module that controls read and write access between the NVS and the software management processor;

a primary current context area in the primary NVS device that stores a current copy of system software, and may be accessed by the SVM to upload the current copy of system software to the software management processor;

a secondary current context area in the secondary NVS device that stores a backup copy of system software, and may be accessed by the SVM to upload the backup copy of system software to the software management processor;

a primary alternate context area in the primary NVS device that may be accessed by the SVM to download a new version of system software through the software management processor;

a secondary alternate context area in the secondary NVS device that may be accessed by the SVM to create a backup copy of the new version of system software in the primary alternate context area;

means of exchanging the system software in the primary current context area with the system software in the primary alternate context area, and exchanging the backup system software in the secondary current context area with the backup system software in the secondary alternate context area; and

a plurality of processor modules coupled to the software management processor that retrieve system software from the software management processor.

72. A method of managing system software in a multiprocessor system having a primary and a secondary non-volatile storage (NVS) device and a plurality of processor modules, comprising the steps of:

enabling a user to download a new version of system software from a file server to a primary alternate context area of the primary NVS device, and creating a backup copy of the new version of system software in a secondary alternate context area of the secondary NVS device;

enabling a user to exchange the new version of system software stored in the primary alternate context area with a current version of system software stored in a primary current context area of the primary NVS device, and exchange the backup copy of the new version of system software with a backup copy of the current version of system software stored in a secondary current context area;

uploading the current copy of system software from the primary NVS device to a software management processor upon initialization of the multiprocessor system; and

loading the current copy of system software through the software management processor to the plurality of processor modules.